

## PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2000-330794  
(43)Date of publication of application : 30.11.2000

(51)Int.Cl.

606F 9/45

(21)Application number : 11-169869  
(22)Date of filing : 16.06.1999

(71)Applicant : NEC CORP  
(72)Inventor : HATAKEYAMA AKEMI

(30)Priority

Priority number : 11070434 Priority date : 16.03.1999 Priority country : JP

## (54) METHOD FOR CONVERTING MHEG DOCUMENT INTO HTML DOCUMENT

## (57)Abstract:

PROBLEM TO BE SOLVED: To convert an MHEG object which can not be converted as a component by converting an MHEG document into an HTML document which has a description of an object, and a description and a script of CSS.

SOLUTION: The shape and operation of a slider object and a link object are extracted from a document described in MHEG and decomposed. The shape of the slider object is converted into the properties of multiple objects described in HTML. Other properties of the slider object are converted into variables described by a script of HTML. Link effect regarding the slider object described in the link object is converted into a method described by a script of HTML. The event type of the link object is converted into the event handler of an object of HTML. The shape and operation represented in the generated HTML are related.



(19)日本国特許庁 (JP)

## (12) 公開特許公報 (A)

(11)特許出願番号

特開2000-330794

(P2000-330794A)

(43)公開日 平成12年11月30日 (2000.11.30)

(51)Int.Cl.  
G 0 6 F 9/45

識別記号

P I  
G 0 6 F 9/44チ-コ-ト (参考)  
3 2 2 Z 5 B 0 8 1

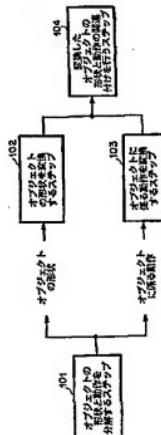
		審査請求 有 請求項の数20 O.L (全 27 頁)
(21)出願番号	特願平11-169869	(71)出願人 000004237 日本電気株式会社 東京都港区芝五丁目7番1号
(22)出願日	平成11年6月16日(1999.6.16)	(72)発明者 岩山 朱美 東京都港区芝五丁目7番1号 日本電気株 式会社内
(31)優先権主張番号	特願平11-70434	(74)代理人 100065385 弁理士 山下 稔平
(32)優先日	平成11年3月16日(1999.3.16)	F ターム(参考) 5B081 AA10 B308
(33)優先権主張国	日本 (JP)	

(54)【発明の名称】 MHEG文書をHTML文書に変換する方法

## (57)【要約】

【課題】 部品として変換できないMHEGオブジェクトを変換できるMHEG文書からHTML文書への変換方法を提供する。

【解決手段】 MHEG文書中の第1MHEGオブジェクトのプロパティをHTML文書上の1以上のオブジェクトのプロパティに変換し、第1MHEGオブジェクトのHTML文書上のオブジェクトのプロパティに変換されるプロパティ以外のプロパティをHTML文書上のスクリプトで記述される変数に変換し、リンクオブジェクトのイベントタイプをリンクオブジェクトのイベントソースに対応したHTML文書上の第2オブジェクトに関連付けられたイベントハンドラに変換し、第2MHEGオブジェクトをイベントソースとして持つリンクオブジェクトのリンクイフェクトをイベントハンドラと関連づけられ且つスクリプトにより記述されるメソッドに変換する。



## 【特許請求の範囲】

【請求項1】 1又は2以上のMHEG(Multimedia and Hypermedia Expert Group)オブジェクトを有するMHEG文書中、1又は2以上のオブジェクトの記述、CSS(Cascading Style Sheet)の記述及びスクリプトを有するHTML(Hypertext Markup Language)文書に変換することを特徴とするMHEG文書をHTML文書に変換する方法。

【請求項2】 前記HTML文書のオブジェクトにはプラグインオブジェクトが含まれることを特徴とする請求項1に記載のMHEG文書をHTML文書に変換する方法。

【請求項3】 MHEG(Multimedia and Hypermedia Expert Group)文書中の第1のMHEGオブジェクトのプロパティの少なくとも一部をHTML(Hypertext Markup Language)文書上の1又は2以上のオブジェクトのプロパティに変換するステップと、

前記第1のMHEGオブジェクトの前記HTML文書上のオブジェクトのプロパティに変換されるプロパティ以外のプロパティの少なくとも一部をHTML文書上のスクリプトにより記述される変数に変換するステップと、第2のMHEGオブジェクトをイベントソースとして持つリンクオブジェクトのイベントタイプを該リンクオブジェクトの前記イベントソースに対応したHTML文書上のオブジェクトに関連付けられたイベントハンドラに変換するステップと、

前記第2のMHEGオブジェクトをイベントソースとして持つ前記リンクオブジェクトのリンクifiketを前記イベントハンドラと関連づけられ且つ前記スクリプトにより記述されるメソッド又はメソッドの組み合わせに変換するステップと、

を有し、前記第1のMHEGオブジェクトと前記第2のMHEGオブジェクトは、同一であるか又は異なることを特徴とするMHEG文書をHTML文書に変換する方法。

【請求項4】 前記HTML文書上のオブジェクトのプロパティに変換されるプロパティは前記第1のMHEGオブジェクトの形状プロパティの少なくとも一部を含むことを特徴とする請求項3に記載のMHEG文書をHTML文書に変換する方法。

【請求項5】 前記メソッド又はメソッドの組み合わせは、前記第1のMHEGオブジェクトのプロパティにより変換された前記HTML文書上のオブジェクトのプロパティを扱うことを特徴とする請求項3又は4に記載のMHEG文書をHTML文書に変換する方法。

【請求項6】 前記メソッド又はメソッドの組み合わせは、前記第1のMHEGオブジェクトのプロパティにより変換された前記HTML文書上のオブジェクトのプロパティを取得することを特徴とする請求項5に記載のMHEG文書をHTML文書に変換する方法。

【請求項7】 前記メソッド又はメソッドの組み合わせは、前記第1のMHEGオブジェクトのプロパティにより変換された前記HTML文書上のオブジェクトのプロパティを変化させることを特徴とする請求項5に記載のMHEG文書をHTML文書に変換する方法。

【請求項8】 前記第1のMHEGオブジェクトのプロパティにより変換されるプロパティは、一对のDIVタグ及び該一对のDIVタグに挟まれるスタイル属性の記述文であることを特徴とする請求項3乃至7のいずれか1項に記載のMHEG文書をHTML文書に変換する方法。

【請求項9】 前記スクリプト内で同一種類の複数のMHEGオブジェクトより変換されたHTML文書上のオブジェクトを配列で管理することを特徴とする請求項3乃至8のいずれか1項に記載のMHEG文書をHTML文書に変換する方法。

【請求項10】 前記スクリプトはジャバスクリプト(JavaScript)であることを特徴とする請求項3乃至9のいずれか1項に記載のMHEG文書をHTML文書に変換する方法。

【請求項11】 前記第1のMHEGオブジェクトはスタイルオブジェクトであることを特徴とする請求項3乃至10のいずれか1項に記載のMHEG文書をHTML文書に変換する方法。

【請求項12】 前記第1及び第2のMHEGオブジェクトはパレットオブジェクトではなく、前記MHEG文書中の前記パレットオブジェクトをカラーのスタイル属性に変換するステップを有することを特徴とする請求項3乃至11のいずれか1項に記載のMHEG文書をHTML文書に変換する方法。

【請求項13】 前記第1及び第2のMHEGオブジェクトは変数オブジェクトではなく、MHEG文書中の前記変数オブジェクトをスクリプトの変数に変換するステップを有することを特徴とする請求項3乃至11のいずれか1項に記載のMHEG文書をHTML文書に変換する方法。

【請求項14】 MHEG文書中の前記第1又は第2のオブジェクト又はその他のMHEGオブジェクトの少なくとも一部をHTML文書上のプラグインオブジェクトに変換するステップを有することを特徴とする請求項3乃至11のいずれか1項に記載のMHEG文書をHTML文書に変換する方法。

【請求項15】 MHEG文書中の前記第1又は第2のオブジェクト又はその他のMHEGオブジェクトの少なくとも一部をHTML文書上のスクリプトに変換するステップを有することを特徴とする請求項1乃至11のいずれか1項に記載のMHEG文書をHTML文書に変換する方法。

【請求項16】 MHEG文書中のパレットオブジェクトをカラーのスタイル属性に変換するステップを有する

ことを特徴とするMHEG文書をHTML文書に変換する方法。

【請求項17】 MHEG文書中の変数オブジェクトをスクリプトの変数に変換するステップを有することを特徴とするMHEG文書をHTML文書に変換する方法。  
【請求項18】 MHEG文書中のMHEGオブジェクトの少なくとも一部をHTML文書上の埋め込みオブジェクトに変換するステップを有することを特徴とするMHEG文書をHTML文書に変換する方法。

【請求項19】 MHEG文書中のMHEGオブジェクトの少なくとも一部をHTML文書上のスクリプトに変換するステップを有することを特徴とするMHEG文書をHTML文書に変換する方法。

【請求項20】 請求項1乃至19のいずれか1項に記載のMHTG文書をHTML文書に変換する方法をコンピュータに実行させるためのプログラムを記録したことを特徴とするコンピュータ読み取り可能な記録媒体。

#### 【発明の詳細な説明】

##### 【0001】

【発明の属する技術分野】 本発明は、MHEG (Multimedia and Hypermedia Expert Group) 文書をHTML (HyperText Markup Language) 文書に変換する方法に関し、特に、HTML文書上のオブジェクトに直接変換することができないMHEGオブジェクトを含むMHEG文書をHTML文書に変換する方法に関する。

##### 【0002】

【従来の技術】 デジタル放送の開始に合わせて、データ放送も順次始まりつつある。データ放送の符号化方式として、或いは、セットトップユニットのハイレベルのAPI (Application Programming Interface) の標準として、DAVIC (Digital Audio-Visual Council) がMHEG-5を採用した。MHEGは、マルチメディア情報のオブジェクトやハイバースペース情報のオブジェクトの表現 (representation) を交換 (interchange) するための標準であり、ISO/IEC (International Organization for Standardization/International Electrotechnical Commission) 及びITU (International Telecommunication Union) の標準である。MHEGのモデルはクラス (「Ingredient Class」という。) より生成されるインスタンスとしてのオブジェクト (MHEGオブジェクト、「Ingredient」という。) として記述される。MHEG文書は、原始的にはISOの規格であるASN1 (Abstract Syntax Notation version 1) により記述され、バイナリコードに変換される。

【0003】一方で、インターネットでは、ホームページ記述言語としてはHTML 4.0が標準となってきている。

【0004】これまで、MHEG文書とHTML文書は別々に使用されることが想定されていたが、WWW (for Wide Web) の表現能力が高まり、それぞれのコンテンツを相互利用するニーズが高まってきた。

##### 【0005】

【発明が解決しようとする課題】 しかしながら、MHEG-5とHTML 4.0、CSS (Cascading Style Sheet)、JavaScript (JavaScript) の組み合わせとを比較すると機能的には似ているが、部品として変換できないMHEGオブジェクトがあり、今後のインターネットと放送のコンテンツ相互利用を考えるとそのような機能の変換が必須である。

【0006】そこで、本発明は、部品として変換することができないMHEGオブジェクトを変換することができるMHEG文書をHTML文書に変換する方法を提供することを目的とする。

【0007】また、本発明は、部品として変換することができないスライダオブジェクトを変換することができるMHEG文書をHTML文書に変換する方法を提供することを目的とする。

##### 【0008】

【課題を解決するための手段】 本発明によるMHEG文書をHTML文書に変換する方法は、1又は2以上のMHEGオブジェクトを有するMHEG文書中を、1又は2以上のオブジェクトの記述、CSSの記述及びスクリプトを有するHTML文書に変換することを特徴とする。

【0009】また、本発明によるMHEG文書をHTML文書に変換する方法は、上記のMHEG文書をHTML文書に変換する方法において、前記HTML文書のオブジェクトには埋め込まれることを特徴とする。更に、本発明によるMHEG文書をHTML文書に変換する方法は、MHEG文書中の第1のMHEGオブジェクトのプロパティの少なくとも一部をHTML文書上の1又は2以上のオブジェクトのプロパティに変換するステップと、前記第1のMHEGオブジェクトの前記HTML文書上のオブジェクトのプロパティに変換されるプロパティ以外のプロパティの少なくとも一部をHTML文書上のスクリプトにより記述される変数に変換するステップと、第2のMHEGオブジェクトをイベントソースとして持つリンクオブジェクトのイベントソースと前記第1のMHEGオブジェクトのイベントソースに対応したHTML文書上のオブジェクトに関連付けられたイベントハンドラに変換するステップと、前記第2のMHEGオブジェクトをイベントソースとして持つ前記リンクオブジェクトのリンクイフェクトを前記イベントハンドラと関連づけられ且つ前記スクリプトにより記述されるメソッド又はメソッドの組み合わせに変換するステップと、を有し、前記第1のMHEGオブジェクトと前記第2のMHEGオブジェクトは、同一であるか又は異なることを特徴とする。

【0010】更に、本発明によるMHEG文書をHTML文書に変換する方法は、上記のMHEG文書をHTML文書に変換する方法において、前記HTML文書上の

オブジェクトのプロパティに変換されるプロパティは前記第1のMHEGオブジェクトの形状プロパティの少なくとも一部を含むことを特徴とする。

【0011】更に、本発明によるMHEG文書をHTML文書に変換する方法は、上記のMHEG文書をHTML文書に変換する方法において、前記メソッド又はメソッドの組み合わせは、前記第1のMHEGオブジェクトのプロパティより変換された前記HTML文書上のオブジェクトのプロパティを扱うことを特徴とする。

【0012】更に、本発明によるMHEG文書をHTML文書に変換する方法は、上記のMHEG文書をHTML文書に変換する方法において、前記メソッド又はメソッドの組み合わせは、前記第1のMHEGオブジェクトのプロパティより変換された前記HTML文書上のオブジェクトのプロパティを取得することを特徴とする。

【0013】更に、本発明によるMHEG文書をHTML文書に変換する方法は、上記のMHEG文書をHTML文書に変換する方法において、前記メソッド又はメソッドの組み合わせは、前記第1のMHEGオブジェクトのプロパティより変換された前記HTML文書上のオブジェクトのプロパティを変化させることを特徴とする。

【0014】更に、本発明によるMHEG文書をHTML文書に変換する方法は、上記のMHEG文書をHTML文書に変換する方法において、前記第1のMHEGオブジェクトのプロパティより変換された前記HTML文書上のオブジェクトのプロパティを変化させることを特徴とする。

【0015】更に、本発明によるMHEG文書をHTML文書に変換する方法は、上記のMHEG文書をHTML文書に変換する方法において、前記スクリプト内で同一種類の複数のMHEGオブジェクトにより変換されたHTML文書上のオブジェクトを配列で管理することを特徴とする。

【0016】更に、本発明によるMHEG文書をHTML文書に変換する方法は、上記のMHEG文書をHTML文書に変換する方法において、前記スクリプトはジャバスク립트 (JavaScript) であることを特徴とする。

【0017】更に、本発明によるMHEG文書をHTML文書に変換する方法は、上記のMHEG文書をHTML文書に変換する方法において、前記第1のMHEGオブジェクトはスライダオブジェクトであることを特徴とする。

【0018】更に、本発明によるMHEG文書をHTML文書に変換する方法は、上記のMHEG文書をHTML文書に変換する方法において、前記第1及び第2のMHEGオブジェクトはパレットオブジェクトではなく、前記MHEG文書中の前記パレットオブジェクトをカラーのスタイル属性に変換するステップを有することを特徴とする。

徴とする。

【0019】更に、本発明によるMHEG文書をHTML文書に変換する方法は、上記のMHEG文書をHTML文書に変換する方法において、前記第1及び第2のMHEGオブジェクトは変数オブジェクトではなく、MHEG文書中の前記変数オブジェクトをスクリプトの変数に変換するステップを有することを特徴とする。

【0020】更に、本発明によるMHEG文書をHTML文書に変換する方法は、上記のMHEG文書をHTML文書に変換する方法において、MHEG文書中の前記第1又是第2のオブジェクト又はその他のMHEGオブジェクトの少なくとも一部をHTML文書上のプラグインオブジェクトに変換するステップを有することを特徴とする。更に、本発明によるMHEG文書をHTML文書に変換する方法は、上記のMHEG文書をHTML文書に変換する方法において、MHEG文書中の前記第1又是第2のオブジェクト又はその他のMHEGオブジェクトの少なくとも一部をHTML文書上のスクリプトに変換するステップを有することを特徴とする。

【0021】更に、本発明によるMHEG文書をHTML文書に変換する方法は、MHEG文書中のパレットオブジェクトをカラーのスタイル属性に変換するステップを有することを特徴とする。

【0022】更に、本発明によるMHEG文書をHTML文書に変換する方法は、MHEG文書中の変数オブジェクトをスクリプトの変数に変換するステップを有することを特徴とする。

【0023】更に、本発明によるMHEG文書をHTML文書に変換する方法は、MHEG文書中のMHEGオブジェクトの少なくとも一部をHTML文書上のプラグインオブジェクトに変換するステップを有することを特徴とする。

【0024】更に、本発明によるMHEG文書をHTML文書に変換する方法は、MHEG文書中のMHEGオブジェクトの少なくとも一部をHTML文書上のスクリプトに変換するステップを有することを特徴とする。

【0025】本発明によるコンピュータ読み取り可能な記録媒体は、上記のMHTG文書をHTML文書に変換する方法をコンピュータに実行させるためのプログラムを記録したことを特徴とする。

【0026】

【発明の実施の形態】【実施形態1】本発明の実施形態1においては、特にスライダオブジェクトの変換方法を取り説明する。

【0027】本実施形態によるスライダオブジェクトの変換方法は、スライダオブジェクトを持つマルチメディア記述言語であるMHEGで記述された文書を、スライダオブジェクトを持たない他のマルチメディア記述言語であるHTMLで記述された文書に変換する際の、スライダオブジェクトの変換方法である。本実施形態は、変

換元の言語により記述されるスライダオブジェクトおよびその機能を、変換先の言語により記述される他のオブジェクトおよび機能の組み合わせにより実現し、変換先の言語にスライダオブジェクトがなくても動作に支障のないよう変換できることを特徴とする。

【0028】本発明の実施形態として、MHEG-5により記述されたスライダオブジェクトからHTML 4.0、CSS、JavaScriptの組み合わせ（以下、「HTML 4.0 set」という。）への変換を示す。

【0029】以下、本発明の構成を図面を参照して説明する。

【0030】図1は、本発明の実施形態によるスライダオブジェクトの変換方法を示すブロック図である。図1において、101はMHEG-5で記述されたアプリケーションおよびシーンの中からスライダオブジェクトの形状と動作、及びスライダオブジェクトへのアクションを持っているLinkオブジェクト（リンクオブジェクト）の抽出と分解を行うステップである。102はスライダオブジェクトのプロパティの一部である形状を変換するステップであり、スライダオブジェクトの形状をHTML 4.0 setにより記述される複数のオブジェクトのプロパティに変換する。また、102では、スライダオブジェクトの他のプロパティを、HTML 4.0 set上上のJavaScriptにより記述される変数に変換する。103はLinkオブジェクト内に記述されたスライダオブジェクトに係るリンクイフェクトをHTML 4.0 set上上のJavaScriptにより記述されるメソッド又はメソッドの組み合わせに変換するステップである。また、このステップで、Linkオブジェクトのイベントタイプは当該Linkオブジェクトのイベントソースに対応するHTML 4.0 set上上のオブジェクトのイベントハンドラに変換される。104は、102および103により生成されたHTML 4.0 set上で表現された形状および動作の関連付けを行うステップである。このステップで、メソッド又はメソッドの組み合わせは、リンクオブジェクトのイベントタイプに対応するHTML 4.0 set上のイベントハンドラを扱うこととなる。

【0031】次に、本発明の実施形態の動作について説明する。

【0032】本実施形態ではMHEG-5のアプリケーションおよびシーンをHTML 4.0 setの1ファイル（ページ）に変換することを想定している。

【0033】本発明の実施形態のうち、スライダオブジェクトの変換については、図2に示すようにスライダをインジケーターと持つ2つの部品の組み合わせとして変換する。具体的には各部品を矩形として表現、DIVタグであらわす。また、MHEG-5記述における属性値からHTML 4.0 setにおける属性値への変換は図3に示すルールにとづいて行う。図3の属性欄にはMHEG-5のスライダオブジェクトが持つる全ての属性値を示した。種別の欄がM（マンダトリ（Mandatory）の略。）である属性は、必

須の属性であり、O（オプション(option)）を意味する。）である属性は、記述しなくても良い属性である。Oに続くかっこ内の内容は、デフォルト値である。HTML 4.0 set表現の欄は、MHEG-5の各属性のHTML 4.0 setではどのような記述に変換されるかを示している。表現はプロパティである。このプロパティは、styleが先頭に付き、CSSで用いられるスタイル属性である。また、備考欄にはそれらがどのような役割を担うかを示した。HTML 4.0 setの表現欄において“—”は変換には使用しないことを示し、“※”は直接的な変換ができないことを示す。

【0034】以下に各属性値の変換の詳細について述べる。

【0035】Sharedは、MHEG-5アプリケーションの全てのシーンにおいて、このオブジェクトを共通使用するか否かを示す属性である。もし、Sharedがtrueであるならば、各シーンに対応するすべてのHTML 4.0 setファイル中にこのスライダオブジェクトに対応する2つのDIVタグを記述する。OriginalPaletteRefは、カラー属性（HighlightRefColor、SliderRefColor）に対するカラー指定をパレット参照により行う場合に、表記しなければならない属性である。もし、パレットを使わざつ直接的に、たとえばRGBカラーで指定するのであれば、必要な属性である。Orientation属性は、スライダの向き、方向を示す属性である。取りうる値としてup/down/left/rightがあり、MinValueからMaxValueに向かう方向を示している。

【0036】ObjectIdentifierはIDに変換される。仮に、ObjectIdentifier=1000だった場合、スライド部分をID=1000s、外枠部分をID=1000oのように変換する。

【0037】InitiallyActiveはデフォルトの可視性を示す。trueならばstyle.visibility="visible"、falseならばstyle.visibility="hidden"とする。

【0038】OriginalBoxSizeは外枠部分の大きさを示し、OriginalPositionは左の位置を示すので、それぞれstyle.width、style.heightとstyle.left、style.topに変換する。

【0039】HighlightRefColor、SliderRefColorはそれぞれハイライト色、スライダー色を示し、パレット参照で指定する場合と、直接値が入る場合がある。パレット指定している場合は、直接的な指定に変換し、HTML 4.0 setでは\*.color="#00FFFFのような指定に変換する。

【0040】MinValue、MaxValueはOriginalBoxSizeで示された範囲の端点を示す。このとき、Orientationで指定されている方向指定にのっとって、MinとMaxに対応する端辺を決める。

【0041】InitialValueは初期状態でのスライダの相対的な位置であり、MinValue <= InitialValue <= MaxValueの関係にある。

**【0042】**SliderStyleはNormal/Thermometer/Proportionalの3種類をとりうる。各スタイルの概要例を、Normalは図4(a)に、Thermometerは図4(b)に、Proportionalは図4(c)に示した。

**【0043】**InitialPortionはOrientation = proportionalの場合のみに有効な属性であり、スライド部分の大さき(幅)を指定する。

**【0044】**これらの属性を用いて図5(a)に示すスライダオブジェクトを、HTML 4.0 setに変換した結果は、図5(b)のようになる。

**【0045】**次に、スライダオブジェクトに係るリンクイフェクトの要素であるアクションの変換方法を示す。図6には各アクションのMHEG-5での記述形式、HTML 4.0 set[JavaScript記述]に変換した場合の関数の引数概要、各アクションの動作を示した。それぞれのJavaScriptで記述された関数の詳細を図7の(x)、(a)と図8の(b)～(e)に示す。なお、各関数の中で使用されている変数のうち、StepSizeとOrientationとSliderTypeはページ読み込み時にJavaScriptにより記述される変数として図7(x)の様に設定する。

**【0046】**以下に関数の詳細説明を示す。

**【0047】**図7(x)のInitialzeで定義される変数はMHEG-5のスライダの属性値であって、HTML 4.0 set style指定では示せない属性に対応する。図7(a)のStep(IDs、ステップ数)では、ID=IDsのスライド部分に対応するDIVタグの位置をステップ数だけ変更する。図8(b)のSetSliderValue(IDs、新しい値)は、ID=IDsのスライド部分に対応するDIVタグの位置を「新しい値」に対応する位置に変更する。図8(c)のGetSliderValue(IDs)は、ID=IDsのスライド部分に対応するDIVタグの位置から、スライダの値を判定し返す。図8(d)のSetPortion(IDs、新しい値)はスライダType:Proportionalの場合のみ有効であり、ID=IDsに対応するインジケータの可動方向軸に対する表示範囲の大きさを変更する。図8(e)のGetPortion(IDs)はスライダType:Proportionalの場合のみ有効でありID=IDsに対応するインジケータの可動方向軸に対する表示範囲の大きさを返す。

**【0048】**図9にスライダオブジェクトおよびスライダオブジェクトに係るアクションを記述したMHEG-5形式の文書の例を示す。また、このMHEG-5形式の文書を変換した結果のHTML 4.0 set文書の例を図10～15に示す。このHTML 4.0 set文書は、実際にブラウザ上で表示できることを確認したものである。サンプル中にBitmapオブジェクトがあるが、BitmapのようなMHEG-5とHTML 4.0 setで1対1に対応するオブジェクトは属性値もほぼ1対1に対応するため、直接的に変換を行った。

**【0049】**図9のMHEG-5文書には、横に並ぶ3つのスライダが記述されている。左端のスライダのタイプはNormalであり、真ん中のスライダのタイプはThermometerであり、右端のスライダのタイプはProportionalであ

る。また、左端のスライダの下に配置されるボタンのセットマップも記述されている。Link100には、ピットマップにUserInputのイベントが発生したときに左端のスライダをステップ移動させることが記述されている。Link110には、真ん中のスライダにCursorEnterのイベントが発生したときに、真ん中のスライダの値を右端のスライダのインジケータのPortionに設定することが記述されている。Link120には、右端のスライダにCursorEnterのイベントが発生したときに、左端のスライダのインジケータのPortionの値を真ん中のスライダに設定することが記述されている。

**【0050】**図10～15のHTML 4.0 set文書では、StepSize、Orientation、SliderType、Pointは配列により管理されている。Step、SetSliderValue、GetSliderValue、SetPortion、GetPortionはJavaScriptにより記述されるメソッドであり、MHEG-5文書中のリンクイフェクトの各要素のメソッドに対応している。これらのメソッド中でも、スライダは配列により管理されている。

**【0051】**ID=ID70000、ID=ID70000sのDIVタグは、MHEG-5文書中のSlider7000に対応するものであり、ID=ID70000sは、スライダの枠を表し、ID=ID70000sは、スライダのインジケータを表す。同様に、ID=ID70010、ID=ID70010sのDIVタグは、MHEG-5文書中のSlider7001に対応するものであり、ID=ID70010sは、スライダの枠を表し、ID=ID70010sは、スライダのインジケータを表す。更に、ID=ID70020、ID=ID70020sのDIVタグは、MHEG-5文書中のSlider7002に対応するものであり、ID=ID70020sは、スライダの枠を表し、ID=ID70020sは、スライダのインジケータを表す。

**【0052】**ID=ID70001sのDIVタグのOnMouseOverのイベントハンドラには、SetPortionとGetPortionより成るメソッドの組み合わせが関連づけられている。ID=ID70001sのDIVタグは、MHEG-5文書のLINK110のEventSourceに対応する。OnMouseOverは、MHEG-5文書のLINK110のイベントタイプであるCursorEnterに対応する。メソッドであるSetPortionとGetPortionは、MHEG-5文書のLINK110のLinkEffectに対応する。

**【0053】**ID=ID70020sのDIVタグのOnMouseOverのイベントハンドラには、SetSliderValueとGetPortionより成るメソッドの組み合わせが関連づけられている。ID=ID70020sのDIVタグは、MHEG-5文書のLINK120のEventSourceに対応する。OnMouseOverは、MHEG-5文書のLINK120のイベントタイプであるCursorEnterに対応する。メソッドであるSetSliderValueとGetPortionは、MHEG-5文書のLINK120のLinkEffectに対応する。

**【0054】**SRC="button.bmp"のIMGタグのOnClickのイベントハンドラには、メソッドであるStepが関連づけられている。SRC="button.bmp"のIMGタグはMHEG-5文書のLINK100のEventSourceに対応する。OnClickは、MHEG-5文書のLINK100のEventTypeに対応する。メソッドであるSt

epはMHEG-5文書のLINK100のLinkEffectに対応する。

【0055】なお、上記の図10～15の説明では、ID=7001sのD1Vタグ節にOnMouseOverのイベントハンドラを、ID=7002sのD1Vタグ節にOnMouseOverのイベントハンドラを、<IMG SRC="button.bm" . . . />の節にonClickのイベントハンドラを挿入したが、イベントハンドラの挿入位置はD1Vタグ節、IMG節に限られるものではなく、OBJECT節その他のタグ節でもよい。これらのタグ節を基にブラウザによりHTML上へのオブジェクトが生成される。

【0056】また、上記の実施形態では、スライダオブジェクトの形状を2つのD1Vタグ節に変換したが、スライダオブジェクトのタイプが、ノーマルである場合には、スライダオブジェクトの形状をインジケーターに対応したD1Vタグ節、枠部分の端からインジケーターまでの部分に対応したD1Vタグ節、及び枠部分の他端からインジケーターまでの部分に対応したD1Vタグ節に変換して、後2者のonClickイベントハンドラにSTEPのメソッドを対応付けることができる。

【0057】スライダオブジェクトに関しては、1つのスライダオブジェクトからHTML 4.0 set上の複数のオブジェクトを生成するとしたが、MHEG文書上のオブジェクトの種類に応じて、MHEG-5文書上のオブジェクトから1つのHTML 4.0 set上のオブジェクトを生成しても良い。

【0058】【実施形態2】次に、本発明の実施形態2について、図面を参照して説明する。

【0059】図16乃至21に、本発明の実施形態2による各MHEGオブジェクトの各プロパティの変換表を示す。

【0060】図16乃至21において、左端欄はMHEGオブジェクト、中欄はMHEGオブジェクトのプロパティ、右端欄は変換後のHTML 4.0 set上の表現形式を示す。

【0061】CursorShapeオブジェクトは、<OBJECT>タグによって囲まれる記述もしくは、CSSの属性に変換される。<OBJECT>タグによって囲まれる記述に変換される場合は、ObjectIdentifierは、IDに変換される。InitiallyActiveは、スタイル属性のvisibilityに変換される。ContentHookは、スクリプトのCLASSIDに変換される。OrigContentは、PARAMタグに変換される。PARAMタグ内のfilename等は、Cursorの形状を示すピットマップファイル名などである。SharedRefは、変換の処理中にflag情報を持たれ、使用する場所で<OBJECT>タグによって囲まれる記述に変換される。

【0062】Paletteオブジェクトは、HTML 4.0 set上の特定のオブジェクトには変換できず、関連する個々のオブジェクトのスタイル属性であるstyle.colorに変換される。

【0063】Fontオブジェクトは、CursorShapeと同様

に変換される。

【0064】BooleanVariableオブジェクト、IntegerVariableオブジェクト、OctetStringVariableオブジェクト、ObjectRefVariableオブジェクト、ContentRefVariableオブジェクトは、一種の変数オブジェクトであり、そのObjectIdentifierは、スクリプトの変数に変換され、そのOriginalValueは、その変数の値に変換される。

【0065】TokenGroupオブジェクトのTokenGroupItemに於けるオブジェクトに対応したHTML 4.0 set文書上のオブジェクトには、連番IDが付され、そのMovementTableは、SetFocusを用いたスクリプトによるメソッドに変換され、そのメソッドには、TokenGroupオブジェクトの名称であるObjectIdentifierが付される。ListGroupについても同様である。

【0066】DynamicLineArtオブジェクトのOriginalPartRef、BorderedBoundingBox、OriginalLineWidth、OriginalLineStyle、OriginalRefLineColor、OriginalFillColorは、プラグインオブジェクトに変換される。PARAMに横くかっこ内で示されるのは、HTML文書とプラグインオブジェクトとの間の引数である。

【0067】図22乃至24に、本発明の実施形態2によるMHEG各オブジェクトの変換表を示す。アクションは、リンクオブジェクトのリンクイフェクトで使用される。

【0068】AddItemアクションに対応する連番IDはタグであり、AddItemは、スクリプトで記述される。

【0069】Deselectは、OnMouseClickedやOnKeyDownなどのイベントハンドラに関連付けられたスクリプトに変換されるが、オブジェクトにより、イベントハンドラやスクリプトの種類が異なる。

【0070】DrawAreaは、プラグインオブジェクトにより記述され、<OBJECT>で示すのはそのプラグインオブジェクトの種別、<PARAM>で示すのはプラグインオブジェクトと交換する情報である。

【0071】図25に、本発明の実施形態2によるMHEGの各イベントの変換表を示す。イベントは、リンクオブジェクトのイベントタイプに使用される。

【0072】上記の実施形態1、2では、JavaScriptをHTML文書に組み込んでいたが、HTML文書とは別個のファイルに書き込んで、HTML文書から呼び出すようにしても良い。

【0073】また、上記の実施形態1、2においては、スクリプト言語としてJavaScriptを使用したが、VBScriptや Tclなどの他のスクリプト言語を使用することもできる。

【0074】なお、上記の実施形態による方法は、CPUに当該方法を実行させるためのプログラムを記録したCD-ROMなどの記録媒体からCPUが当該プログラムを読み込んで実行することにより実現することもできる。

る。

【0075】また、上記の実施形態による方法は、図2-6に示すように、CPU203に当該方法を実行させるためのプログラム207を記録したCD-ROMなどのプログラム記録媒体206からCPU203が当該プログラム207を読み込んで実行することにより実現することもできる。この際、通常は、CPU203は、オペレーティングシステムにより、当該プログラム207を当該記録媒体206からメインメモリ204に一旦転送した後に、メインメモリ204から当該プログラムを読み込んで実行する。CPUとメインメモリ204を備えるコンピュータは、文書変換装置として機能し、MHEG-5文書201を入力し、これを装置内でHTML 4.0 set文書に変換した後、HTML 4.0 set文書205を出力する。

【0076】また、上記プログラムは、インタプリタ形式の命令を記述したテキスト形式のファイルと、このファイルを読み込んで実行命令に変換するインタプリタより生成されることもある。

【0077】

【発明の効果】以上説明したように、本発明によれば、MHEG文書からHTML文書への変換において、部品として変換することができないMHEGオブジェクトを変換することができる。

【図面の簡単な説明】

【図1】本発明の実施形態による文書変換方法における、オブジェクトを変換する方法を説明するための概念図である。

【図2】本発明の実施形態1で扱うスライダの形状を示す図である。

【図3】本発明の実施形態1によるスライダの属性の変換表である。

【図4】本発明の実施形態1で扱うスライダの種類を表す図である。

【図5】MHEG-5により記述されるスライダと、これを基に本発明の実施形態1による方法により変換したHTML 4.0 setで記述されるスライダの形状を示す図である。

【図6】本発明の実施形態1によるアクションの変換表である。

【図7】本発明の実施形態1による方法により生成されるJavaScriptにより記述される変数とスクリプトを示す図である。

【図8】本発明の実施形態1による方法により生成されるJavaScriptにより記述されるスクリプトを示す図である。

【図9】本発明の実施形態1による方法を適用するMHEG-5文書の例を示す図である。

【図10】図9に示すMHEG-5文書を本発明の実施形態1による方法で変換したHTML 4.0 set文書を示す図の第1部分である。

【図11】図9に示すMHEG-5文書を本発明の実施形態1

による方法で変換したHTML 4.0 set文書を示す図の第2部分である。

【図12】図9に示すMHEG-5文書を本発明の実施形態1による方法で変換したHTML 4.0 set文書を示す図の第3部分である。

【図13】図9に示すMHEG-5文書を本発明の実施形態1による方法で変換したHTML 4.0 set文書を示す図の第4部分である。

【図14】図9に示すMHEG-5文書を本発明の実施形態1による方法で変換したHTML 4.0 set文書を示す図の第5部分である。

【図15】図9に示すMHEG-5文書を本発明の実施形態1による方法で変換したHTML 4.0 set文書を示す図の第6部分である。

【図16】本発明の実施形態2による各MHEGオブジェクトの各プロパティの変換表の第1部分である。

【図17】本発明の実施形態2による各MHEGオブジェクトの各プロパティの変換表の第2部分である。

【図18】本発明の実施形態2による各MHEGオブジェクトの各プロパティの変換表の第3部分である。

【図19】本発明の実施形態2による各MHEGオブジェクトの各プロパティの変換表の第4部分である。

【図20】本発明の実施形態2による各MHEGオブジェクトの各プロパティの変換表の第5部分である。

【図21】本発明の実施形態2による各MHEGオブジェクトの各プロパティの変換表の第6部分である。

【図22】本発明の実施形態2によるMHEGの各アクションの変換表の第1部分である。

【図23】本発明の実施形態2によるMHEGの各アクションの変換表の第2部分である。

【図24】本発明の実施形態2によるMHEGの各アクションの変換表の第3部分である。

【図25】本発明の実施形態2によるMHEGの各イベントの変換表である。

【図26】本発明の実施形態による方法を実現するための装置を示すブロック図である。

【符号の説明】

101 オブジェクトの形状と動作を分解するステップ

102 オブジェクトの形状を変換するステップ

103 オブジェクトに係る動作を変換するステップ

104 変換したオブジェクトの形状と動作の関連付けを行なうステップ

201 MHEG-5文書

202 文書変換装置

203 CPU

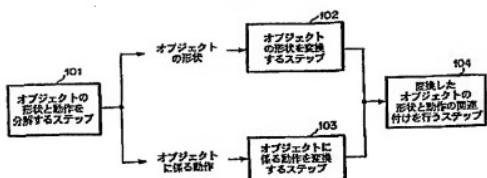
204 メインメモリ

205 HTML 4.0 set文書

206 プログラム記録媒体

207 プログラム

【図1】



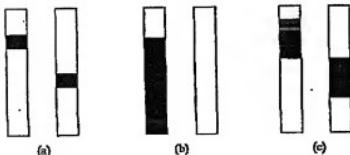
【図2】



【図3】

MHEG-5	属性	表現	HTML 4.0 net	備考
ObjectIdentifier	M	ID	オブジェクトID(後、スタイル部分に適用)	
InitiallyActive	O(True)	style.visibility	可視性の指定	
Shared	O(False)	※	複数のオブジェクトで使用する場合にtrue	
OrignateSize	M	stylewidth, styleheight	オブジェクトの大きさ	
OrignatePosition	O(0, 0)	stylewidth, styleheight	オブジェクトの位置	
OrignateHorizontal	O(HCenter)	※	パラメータ使用する場合は使用されている	
OrignateVertical	O(True)	—	パラメータの値	
HeightWidthColor	O(Transparentクラス使用)	style.borderColor	スタイルの色	
Orientation	M	style	スタイルの適用方向	
InitialValue	O(MValue)	stylewidth, styleheight	スタイル部分の位置	
MaxValue	M	stylewidth, styleheight	スタイル部分の大きさ	
MinValue	M	stylewidth, styleheight	スタイル部分の大きさ	
StepSize	M	stylewidth, styleheight	スタイル部分の大きさ	
SliderType	O(normal)	stylewidth, styleheight	スライダの初期位置	
SliderColor	O(Transparentクラス使用)	style.color	スタイル部分の色	

【図4】



【図6】

MHEG-5	属性	JavaScript表現	HTML 4.0 net	備考
アクション				
Step	O(step(234, 2))	SetVal(ID, step(234, 2))	ステップ表示(234)で2で割った残り(2)を変化	
SetValue	O(setSliderValue(234, 20))	SetVal(ID, setSliderValue(234, 20))	スライダの初期位置を234に変更	
GetSliderValue	O(getSliderValue(1, 1))	GetVal(ID, getSliderValue(1, 1))	スライダの初期位置を1に変更	
SetPortion	O(setPortion(234, 5))	SetVal(ID, setPortion(234, 5))	インジケータの大きさを変更(Proportionalの場合のみ)	
GetPortion	O(getPortion(234, 1))	GetVal(ID, getPortion(234, 1))	インジケータの大きさを変更(Proportionalの場合のみ)	

【図5】

```
{Slider 7901
:InitiallyActive true
:OriginalBoxSize (10 110)
:OriginalPosition (20 20)
:OriginalPaletteRef 1
:EngineResp true
:HighlightColour #FFFFFF
:Orientation left
:InitialValue 50
:MinValue 0
:MaxValue 100
:InitialPortion 20
:StepSize 5
:SliderStyle Proportional
:SliderRefColour #FF0000
}
```

(a)

```
<DIV
ID=7901o
style="visibility: true;
width: 10; height: 100;
left: 10; top: 10;
border:#FFFFFF;">
</DIV>
<DIV
ID=7901s
style="visibility: true;
width: 10; height: 20;
left: 20; top: 20+50-0.5*20;
border:#FFFFFF;">
</DIV>
```

(b)

【図15】

```
ID=ID7002s
style="visibility: visible;
position: absolute;
width: 20; height: 40;
left: 210; right: 240; top: 50; bottom: 70;
background: #0000FF; z-index: 1"
onmouseover
="SetSliderValue(7001, ID7001s, ID7001o, GetPortion(7002, ID7002s))"> </DIV>
<IMG
SRC="button.bmo"
ID=ID8000
style="visibility: visible;
position: absolute;
width: 40; height: 20;
left: 0; top: 230;
onClick="Step(7000, ID7000s, 2)" />
</BODY>
</HTML>
```

【図7】

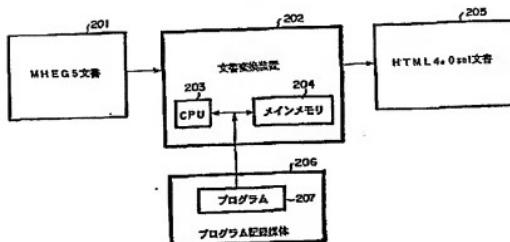
```
function Initialize() {
    var StepSize = StepSize;
    var Orientation = Orientation;
    var SliderType = SliderType;
}
```

(x)

```
function Step(IDs, ステップ数) {
    var stepnum = ステップ数;
    switch(Orientation) {
        case left:
            IDs.style.left = IDs.style.left - StepSize * stepnum;
            break;
        case right:
            IDs.style.left = IDs.style.left + StepSize * stepnum;
            break;
        case up:
            IDs.style.top = IDs.style.top - StepSize * stepnum;
            break;
        case down:
            IDs.style.top = IDs.style.top + StepSize * stepnum;
            break;
    }
}
```

(a)

【図26】



【図8】

```
function SetSliderValue(IDs, 新しい値){
    var newval = 新しい値
    switch(Orientation){
        case left:
            IDs.style.left = IDo.style.right - newval
            break
        case right:
            IDs.style.left = IDo.style.left + newval
            break
        case up:
            IDs.top = IDo.style.bottom - newval
            break
        case down:
            IDs.top = IDo.style.top + newval
            break;
    }
}
```

(b)

```
function GetSliderValue(IDs){
    switch(Orientation){
        case left:
            return IDs.right - IDs.left
            break
        case right:
            return IDs.left - IDo.left
            break
        case up:
            return IDs.bottom - IDs.top
            break
        case down:
            return IDs.top - IDo.top
            break
    }
}
```

(c)

```
function SetPosition(IDs, 新しい値){
    var PosSize = 新しい値
    switch(Orientation){
        case left:
        case right:
            var now = IDs.right - IDs.left
            var d = now - PosSize/2
            IDs.left = IDs.left - d
            IDs.right = IDs.right + d
            break
        case up:
        case down:
            var now = IDs.right - IDs.left
            var d = now - PosSize/2
            IDs.left = IDs.left - d
            IDs.right = IDs.right + d
            break
    }
}
```

(d)

```
function GetPosition(IDs){
    switch(Orientation){
        case left:
        case right:
            return IDs.left - IDs.right
            break
        case up:
        case down:
            return IDs.bottom - IDs.top
            break
    }
}
```

(e)

【図9】

```

{Scene ("~/Scene1" 0)
:stdVersion 1
:items (
  {Slider 7000
    :InitiallyActive true
    :OriginalBoxSize (10 100)
    :OriginalPosition (10 10)
    :OriginalPaletteRef 1
    :EngineResp true
    :HighlightColour #FFFFFF
    :Orientation up
    :InitialValue 50
    :MinValue 0
    :MaxValue 100
    :StepSize 5
    :SliderStyle Normal
    :SliderRefColour #FF0000
  }
  {Slider 7001
    :InitiallyActive true
    :OriginalBoxSize (10 100)
    :OriginalPosition (10 10)
    :OriginalPaletteRef 1
    :EngineResp true
    :HighlightColour #FFFFFF
    :Orientation up
    :InitialValue 50
    :MinValue 0
    :MaxValue 100
    :StepSize 5
    :SliderStyle Thermometer
    :SliderRefColour #00FFFF
  }
  {Slider 7002
    :InitiallyActive true
    :OriginalBoxSize (10 100)
    :OriginalPosition (210 10)
    :OriginalPaletteRef 1
    :EngineResp true
    :HighlightColour #FFFFFF
    :Orientation up
    :InitialValue 50
    :InitialPortion 20
    :MinValue 0
    :MaxValue 100
    :StepSize 5
    :SliderStyle Proportional
    :SliderRefColour #0000FF
  }
)
{Bitmap 8000
  :InitiallyActive true
  :Check 4
  :OrigContent :ContentRef ("~fbutton.png")
  :OrigBoxSize 50 10
  :OrigPosition 10 130
  :Tiling false
  :Link 100
    :EventSource 8000
    :EventType UserInput
    :EventData 1
    :LinkEffect (
      Step (7000 2))
  :Link 110
    :EventSource 7001
    :EventType CursorEnter
    :LinkEffect (
      GetSliderValue(7001 25)
      :SetPortion (7002:IndirectRef 25))
  :Link 120
    :EventSource 7002
    :EventType CursorEnter
    :LinkEffect (
      GetPortion (7002 25)
      :SetSliderValue (7001:IndirectRef 26))
  :IntegerVar 25
    :OrigValue 0
  |
  :IntegerVar 26
    :OrigValue 0
  }
  :InputEventReg 1
  :SceneCS 610 480
  :MovingCursor true
}

```

[图10]

## 【図11】

```

break
case "right":
tmp = IDs.style.left.toString().substring(0, IDs.style.left.length-2)
IDs.style.left=tmp-(-StepSize[num]*stepnum)
break
case "up":
tmp = IDs.style.top.toString().substring(0, IDs.style.top.length-2)
IDs.style.top = tmp-StepSize[num]*stepnum
break
case "down":
tmp = IDs.style.top.toString().substring(0, IDs.style.top.length-2)
IDs.style.top = tmp-(-StepSize[num]*stepnum)
break
}

}

function SetSliderValue(ID, IDs, IDo, newval) {
for(i = 1; i <= 3; i++) {
if(Poin[i] == ID) {
num = i
break
}
}

switch(Orientation[num]) {
case "left":
IDs.style.left = IDo.style.right.toString().substring(0, IDo.style.right.length-2) - newval
IDs.style.widtht = newval
break
case "right":
IDs.style.right = IDo.style.left.toString().substring(0, IDo.style.left.length-2) + newval
IDs.style.widtht = newval
break
case "up":
IDs.style.top = IDo.style.bottom.toString() - newval
IDs.style.height = newval
break
case "down":
IDs.style.bottom = IDo.style.top.toString() + newval
}
}

```

【図12】

```

    lDs.style.height = newval
    break;
}
}

function GetSliderValue(ID, lDs, lDo) {
  for(i = 1; i <= 3;i++){
    if(Point[i] == ID){
      num = i
      break
    }
  }
  switch(orientation[num]){
    case "left":
      orgt = lDo.style.right.toString().substring(0,lDo.style.right.length-2)
      sift = lDs.style.left.toString().substring(0,lDs.style.left.length-2)
      return (orgt - sift)
    case "right":
      srqt = lDs.style.rgt.toString().substring(0,lDs.style.rgt.length-2)
      olft = lDo.style.left.toString().substring(0,lDo.style.left.length-2)
      return (srqt - olft)
    case "up":
      sbtm = lDs.style.bottom.toString()
      stop = lDs.style.top.toString().substring(0,lDs.style.top.length-2)
      return (sbtm - stop)
    case "down":
      sbtm = lDs.style.bottom.toString().substring(0,lDs.style.bottom.length-2)
      stop = lDo.style.top.toString().substring(0,lDo.style.top.length-2)
      return (sbtm - stop)
  }
}

function SetPortion(ID, lDs, lDo, PortSize){
  for(i = 1; i <= 3;i++){
    if(Point[i] == ID){
      num = i
      break
    }
  }
}

```

【図13】

```

if(SlideType[fun] == "Proportional") {
    switch(Orientation[fun]) {
        case "left":
            case "right":
                now = [Ds.style.width, toString(0, [Ds.style.width, length-2])
                d = now - portSize
                cc = d/2
                [Ds.style.left = [Ds.style.left.substring(0, [Ds.style.left.length-2] - dd
                [Ds.style.left = [Ds.style.left.substring(0, [Ds.style.left.length-2] - d
                [Ds.style.width = [Ds.style.width.substring(0, [Ds.style.width.length-2] - d
                break;
        case "up":
            case "down":
                now = [Ds.style.height, toString(0, [Ds.style.height, length-2])
                d = now - portSize
                dd = d/2
                [Ds.style.top = [Ds.style.top.substring(0, [Ds.style.top.length-2] - dd
                [Ds.style.height = [Ds.style.height.substring(0, [Ds.style.height.length-2] - d
                break;
    }
}

function GetPosition([D, [Ds]) {
    for(i = 1; i < 3; i++) {
        if([D[i]] == [D]) {
            num = i
            break;
        }
    }
    if(SlideType[fun] == "Proportional") {
        switch(Orientation[fun]) {
            case "left":
            case "right":
                return [Ds.style.width, toString(0, [Ds.style.width, length-2])
            case "up":
            case "down":
                return [Ds.style.height, toString(0, [Ds.style.height, length-2])
        }
    }
}

```

## 【図14】

```
</SCRIPT>
</HEAD>

<BODY onLoad="Initialize()">
<DIV
  ID=ID7000a
  style="visibility: visible;
position: absolute;
width: 20; height: 200;
left: 10; right: 40; top: 10; bottom: 210;
border: solid 1px #000000;"> </DIV>
<DIV
  ID=ID7000b
  style="visibility: visible;
position: absolute;
width: 20; height: 5;
left: 10; right: 40; top: 60; bottom: 61;
background: #FF0000;"> </DIV>
<DIV
  ID=ID7001a
  style="visibility: visible;
position: absolute;
width: 20; height: 200;
left: 110; right: 140; top: 10; bottom: 210;
border: solid 1px #000000;"> </DIV>
<DIV
  ID=ID7001b
  style="visibility: visible;
position: absolute;
width: 20; height: 50;
left: 110; right: 140; top: 160; bottom: 210;
background: #00FFFF;
onmouseover="SetPortion(7002, ID7002a, ID7002b, GetValue(7001, ID7001a, ID7001b))"> </DIV>
<DIV
  ID=ID7002a
  style="visibility: visible;
position: absolute;
width: 20; height: 200;
left: 210; right: 240; top: 10; bottom: 210;
border: solid 1px #000000;"> </DIV>
```

【図16】

NAME-S	HTML Object
CursorShape	<OBJECT>
ObjectIdentifier	ID
InitiallyActive	style.visibility
ContentHook	CLASSID
OrigContent	PARAM (filename等)
Shared	1. flag情報として持つ。使用する場所で<OBJECT>に変換 2. 参照しているパレットをrgbパレットに変換
Palette	1. 使用する場所でstyle.color等に変換
ObjectIdentifier	<OBJECT>
InitiallyActive	ID
ContentHook	style.visibility
OrigContent	CLASSID
Shared	1. flag情報として持つ。使用する場所で<OBJECT>に変換
Font	<OBJECT>
ObjectIdentifier	ID
InitiallyActive	style.visibility
ContentHook	CLASSID
OrigContent	PARAM (filename等)
Shared	1. flag情報として持つ。使用する場所で<OBJECT>に変換
BooleanVariable	
ObjectIdentifier	変数名
InitiallyActive	- (最初に定義)
Shared	-
OriginalValue	変数値
IntegerVariable	
ObjectIdentifier	変数名
InitiallyActive	- (最初に定義)
Shared	-
OriginalValue	変数値
OctetStringVariable	
ObjectIdentifier	変数名
InitiallyActive	- (最初に定義)
Shared	-
OriginalValue	変数値
ObjectRefVariable	
ObjectIdentifier	変数名1、変数名2
InitiallyActive	- (最初に定義)
Shared	-
OriginalValue	変数値1、変数値2

【図17】

<b>ContentRefVariable</b>	
ObjectIdentifier	変数名1, 変数名2
InitiallyActive	- (最初に定義)
Shared	-
OriginalValue	変数値1, 変数値2
<b>TokenGroup</b>	
ObjectIdentifier	関数名
InitiallyActive	- (最初に定義)
ContentHook	-
OriginalContent	-
Shared	-
MovementTable	JavaScript
TokenGroupItems	連器ID + style.position
NoTokenActionSlots	-
<b>ListGroup</b>	
ObjectIdentifier	関数名
InitiallyActive	- (最初に定義)
ContentHook	-
OriginalContent	-
Shared	-
MovementTable	JavaScript
TokenGroupItems	連器ID
NoTokenActionSlots	-
Positions	style.position
Wraparound	style.position
MultipleSelection	JavaScript
<b>Bitmap</b>	<OBJECT>
ObjectIdentifier	ID
InitiallyActive	style.visibility
ContentHook	CLASSTID
OrigContent	PARAM(filename等)
Shared	1.flag情報として持つ 2. 使用する場所で<OBJECT>に反映
OriginalBoxSize	style.width, style.height
OriginalPosition	style.left, style.top
OriginalPaletteRef	-
Tiling	複数の<OBJECT>に変換
OrigTransparency	-

【図18】

<b>LineArt</b>	<b>&lt;OBJECT&gt;</b>
ObjectIdentifier	ID
InitiallyActive	style.visibility
ContentHook	CLASSID
OrigContent	PARAM(filename等)
Shared	1.flag情報として持つ 2. 使用する場所で<OBJECT>に変換
OriginalBoxSize	style.width style.height
OriginalPosition	style.left, style.top
OriginalPaletteRef	PARAM(palette等)
BorderedBoundingBox	PARAM(GroupBox等)
OriginalLineWidth	PARAM(width等)
OriginalLineStyle	PARAM(style等)
OriginalRefLineColor	PARAM(color等)
OriginalRefFillColor	PARAM(fcolor等)
	<b>&lt;DIV&gt;</b>
ObjectIdentifier	ID
InitiallyActive	style.visibility
Shared	1.flag情報として持つ 2. 使用する場所で<TABLE>に変換
OriginalBoxSize	style.width style.height
OriginalPosition	style.left, style.top
OriginalPaletteRef	詳細するrgb/パレット
OriginalLineWidth	style.border
OriginalLineStyle	style.border
OriginalRefLineColor	style.bordercolor ← rgb/パレット
OriginalRefFillColor	style.background ← rgb/パレット
	<b>&lt;OBJECT&gt;</b>
ObjectIdentifier	ID
InitiallyActive	style.visibility
Shared	CLASSID
OriginalBoxSize	style.width, style.height
OriginalPosition	style.left, style.top
OriginalPaletteRef	PARAM(palette等)
BorderedBoundingBox	PARAM(GroupBox等)
OriginalLineWidth	PARAM(width等)
OriginalLineStyle	PARAM(style等)
OriginalRefLineColor	PARAM(color等)
OriginalRefFillColor	PARAM(fcolor等)

【図19】

Text	OBJ
ObjectIdentifier	ID
InitiallyActive	style.visibility
ContentHook	-
OriginalContent	テキスト文書
Shared	1. flag情報として持つ 2. 使用する場所でOBJに文書に反映
OriginalBoxSize	style.width, style.height
OriginalPosition	style.left, style.top
OriginalPaletteRef	参照するrgbパレット
OriginalFont	style.fontfamily
FontAttributes	style.fontstyle, style fontsize
TextColour	style.color ← rgbパレット
backGroundColour	style.backeround ← rgvパレット
CharacterSet	-
HorizontalJustification	style.text-align
VerticalJustification	style.vertical-align
LineOrientation	-
StartCorner	-
TextWrapping	-
Scream	OBJECT
ObjectIdentifier	ID
InitiallyActive	style.visibility
ContentHook	CLASSID
OriginalContent	PARAM(filename%)
Shared	1. flag情報として持つ 2. 使用する場所で(OBJECT)に反映
Multiplex	-
Storage	-
Looping	PARAM(looping%)
Audio	OBJECT
ObjectIdentifier	ID
InitiallyActive	style.visibility
ComponentTag	CLASSID
OriginalVolume	PARAM(volume%)
Video	OBJECT
ObjectIdentifier	ID
InitiallyActive	style.visibility
OriginalBoxSize	style.width, style.height
OriginalPosition	style.top, style.left
ComponentTag	CLASSID
Termination	PARAM(termination%)

【図20】

RTGraphics	OBJECT
ObjectIdentifier	ID
InitiallyActive	style.visibility
OriginalBoxSize	style.width, style.height
OriginalPosition	style.top, style.left
ComponentID	CLASSID
Termination	PARAM(termination%)
EntryField	<PORN>
ObjectIdentifier	ID
InitiallyActive	style.visibility
ContentBook	-
OrgContent	VALUE
Shared	I. flag情報として持つ。使用する場所でPORNに変換
OriginalBoxSize	style.width, style.height
OriginalPosition	style.top, style.left
OriginalPaletteRef	参照するrgbパレット
OriginalFont	style.fontfamily
FontAttributes	style.fontstyle, style.fontsize
TextColor	style.color ← rgbパレット
BackGroundColour	style.background ← rgbパレット
CharacterSet	-
HorizontalJustification	style.text-align
VerticalJustification	style.vertical-align
LineOrientation	-
StartCorner	-
TextWrapping	-
EngineResp	-
HighlightRefColour	style.bordercolor
Chartlist	<OPTION>
ObscuredInput	<INPUT TYPE='password'>
MaxLength	MAXLENGTH
Hotspot	<OM>
ObjectIdentifier	ID
InitiallyActive	style.visibility
Shared	I. flag情報として持つ。使用する場所でPORNに変換
OriginalBoxSize	style.width, style.height
OriginalPosition	style.top, style.left
OriginalPaletteRef	参照するrgbパレット
EngineResp	-
HighlightRefColour	style.bordercolor ← rgbパレット
ButtonRefColour	style.color ← rgbパレット

【図21】

PushButton	<code>&lt;FORM type="button"&gt;</code>
ObjectIdentifier	ID
InitiallyActive	-
Shared	L.flag情報として持つ 2. 使用する場所で<FORM>に変換
OriginalBoxSize	style.width, style.height
OriginalPosition	style.top, style.left
OriginalPaletteRef	参照するrgbパレット
EngineResp	-
HighlightRefColour	style.bordercolor ← rgbパレット
ButtonRefColour	style.color ← rgbパレット
OriginalLabel	name
CharacterSet	-

SwitchButton	<code>&lt;FORM&gt; type="radio"</code>
ObjectIdentifier	ID
InitiallyActive	-
Shared	L.flag情報として持つ 2. 使用する場所で<FORM>に変換
OriginalBoxSize	style.width, style.height
OriginalPosition	style.top, style.left
OriginalPaletteRef	参照するrgbパレット
EngineResp	-
HighlightRefColour	style.bordercolor ← rgbパレット
ButtonRefColour	style.color ← rgbパレット
OriginalLabel	name
CharacterSet	-
BulletStyle	-

【図22】

WEG-S	HTML 4.0	GetListSize	VAR
Action	Activate	GetOverwriteMode	-
Add	"i"	GetVersion	style.left, style.top, style.right, style.bottom
AddItem	迷惑ID + JavaScript	GetPosition	style.top, style.left
Append	viz	GetUmlanningStatus	JavaScript + visibility
BringToFront	x-index	GetSelectionStatus	JavaScript + visibility
Call	-	GetSliderValue	style.left, style.top, style.right, style.bottom
CallActionScript	pageX, pageY + ID + JavaScript	GetTextContent	ID, var
Clear	<OBJECT><PARAM>	GetTextData	ID, var
Close	-	GetTokenPosition	pageX, pageY + ID
CloseConnection	-	GetVariable	var
Deactivate	-	GetVolume	<OBJECT><PARAM>
Delete	迷惑ID + JavaScript	Launch	location
DeleteI	OnMouseClick, OnKeyDown等	LockScreen	-
Deschedule	迷惑ID + JavaScript	Modulo	JavaScript
Divide	/	None	ID + JavaScript + style.top等
DrawArc	<OBJECT><PARAM>	MoveTo	ID + JavaScript + style.top等
DrawLine	<OBJECT><PARAM>	Multiply	*#
Drawoval	<OBJECT><PARAM>	OpenConnection	-
DrawPolygon	<OBJECT><PARAM>	Prefetch	-
DrawPolyline	<OBJECT><PARAM>	PutBefore	style.z-index
DrawRectangle	<OBJECT><PARAM>	PutBehind	style.z-index
DrawSector	<OBJECT><PARAM>	Quit	-
Fork	-	ReadPersistent	-
GetAvailabilityStatus	-	Ron	style.visibility
GetBoxSize	style.width, style.height	ScaleBitmap	style.width, style.height
GetCellIndex	迷惑ID + JavaScript	ScaleVideo	style.width, style.height
GetCursorPosition	pageX, pageY	ScrollLeft	迷惑ID + JavaScript
GetEngineSupport	-	Select	OnMouseClick, OnKeyDown等
GetEntryField	-	SelectItem	迷惑ID + JavaScript
GetFillColour	<OBJECT><PARAM>, style.backgroundColor	SendEvent	-
GetFirstItem	迷惑ID + JavaScript	SendToBack	style.z-index
GetHighlightStatus	-	SetBoxSize	style.width, style.height
GetInteractionStatus	-	SetCachePriority	-
GetItemStatus	-	SetCounterEndPosition	<OBJECT><PARAM>
GetLabel	<FORM> type="button"	SetCounterPosition	<OBJECT><PARAM>
GetListAnchorFired	-	SetCounterTrigger	<OBJECT><PARAM>
GetLineColour	<OBJECT><PARAM>, style.borderColor	SetCursorPosition	-
GetLineStyle	<OBJECT><PARAM>, style.borderWidth	SetCursorShape	style.cursor
GetLineWidth	<OBJECT><PARAM>, style.border	SetData	-
GetListitem	迷惑ID + JavaScript		

【図23】

【図24】

SetEntryPoint	-
SetFillColour	(OBJECT)>(PARAM), style.background
SetFirstItem	逆番ID + JavaScript
SetFontRef	style.font, style.font-family
SetHighlightStatus	-
SetInteractionStatus	-
SetLabel	(FORM) type='button'
SetLineColour	(OBJECT)>(PARAM), style.bordercolor
SetLineStyle	(OBJECT)>(PARAM), style.borderstyle
SetLineWidth	(OBJECT)>(PARAM), style.border
SetOverrideNode	-
SetPaletteRef	-
SetPosition	style.left, style.top, style.right, style.bottom
SetPositionn	style.top, style.left
SetSliderValue	style.left, style.top, style.right, style.bottom
SetSpeed	(OBJECT)>(PARAM)
SetTimer	SetTimeout
SetTransparency	-
SetVariable	var
SetVolume	var
Spawn	-
Step	style.left, style.top, style.right, style.bottom
Stop	style.visibility
StorePersistent	-
Subtract	"-"
TestVariable	var
Tessie	逆番ID + JavaScript
ToggleItem	逆番ID + JavaScript
TransitionTo	location
Unload	-
UnlockScreen	-

【図25】

<b>Event</b>	<b>IsAvailable</b>	<b>style.visibility: true</b>
ContentAvailable	style.visibility: true	
IsDeleted	-	
IsRunning	style.visibility: true	
IsStopped	style.visibility: false	
UserInput	OnMouseClick, OnMouseKeyDown等	
AnchorFired	-	
TigerFired	SetTimeout	
AsyncStopped	-	
InteractionCompleted	-	
TokenMovedFrom	OnMouseOver, OnClick等	
TokenMovedTo	OnMouseOver, OnClick等	
StreamEvent	Plug-inによる	
StreamPlaying	Plug-inによる	
StreamStopped	Plug-inによる	
CounterTrigger	Plug-inによる	
HighlightOn	style.border	
HighlightOff	style.border	
CursorEnter	OnMouseOver	
CursorLeave	OnMouseOut	
IsSelected	OnClick, OnKeyDown等	
IsDeselected	OnClick, OnKeyDown等	
TestEvent	OnClick, OnKeyDown等	
FirstItemPresented	style.visibility: true	
LastItemPresented	style.visibility: true	
HeadItems	style.visibility: true	
TailItems	style.visibility: true	
ItemSelected	OnClick, OnKeyDown等	
ItemDeselected	OnClick, OnKeyDown等	
EntryFieldFull	-	